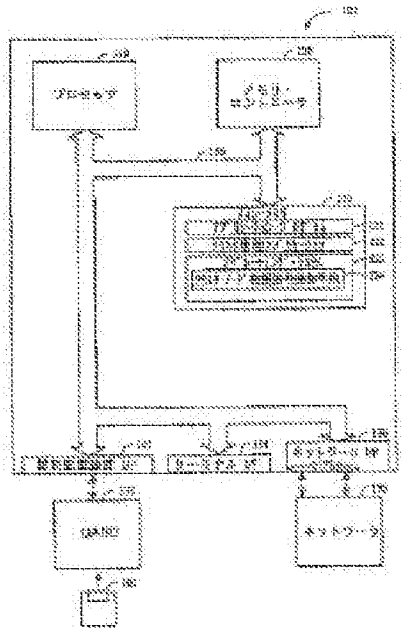
GENERATION SYSTEM FOR CLUSTER NODE RELIEF SIGNAL

Publication number: JP2000293497 (A)
Publication date: 2000-10-20
Inventor(s): BLOCK TIMOTHY R; RODNEY LEE LOVE
Applicant(s): IBM
Classification:
- international: ***G06F15/177; G06F11/00; G06F11/07; G06F11/30; G06F15/16;*** **G06F11/00; G06F11/07; G06F11/30; G06F15/16;** (IPC1-7): G06F15/177; G06F15/16
- European:
Application number: JP20000073269 20000316
Priority number(s): US19990281026 19990330

Also published as:
US6442713 (B1)
TW466399 (B)
SG90111 (A1)
KR20010006847 (A)
CA2290289 (A1)
more >>

Abstract of **JP 2000293497 (A)**

PROBLEM TO BE SOLVED: To improve the reliability of a cluster by sending a previously generated relief message to other nodes of a cluster when a fault phenomenon is detected. SOLUTION: A main memory 120 properly includes one or multiple application programs 121, a cluster management application 122, and an operating system 123 including a cluster-node relief signal generating means 124. This cluster-node relief signal generating means 123 provides a mechanism which needs to send a relief signal to other nodes in a cluster in the case of a serious fault occurring to a node. Namely, the cluster-node relief signal generating means 124 integrated with the operating system 123 includes a relief signal sending-out method, a previously generated relief message, and a dedicated relief signal for sending the message. Then when the fault phenomenon is detected, the previously generated relief message is sent to other nodes of the cluster.



(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号
特開2000－293497
（P2000－293497A）
(43)公開日　平成12年10月20日(2000.10.20)

(51)Int.Cl.[7]		識別記号	FI		テーマコード（参考）
Ｇ０６Ｆ	15/177	６７８	Ｇ０６Ｆ 15/177	６７８Ａ	
	15/16	６４０	15/16	６４０Ａ	

審査請求　有　　請求項の数38　ＯＬ　（全 13 頁）

(21)出願番号　特願2000－73269(P2000－73269)

(22)出願日　平成12年３月16日(2000.3.16)

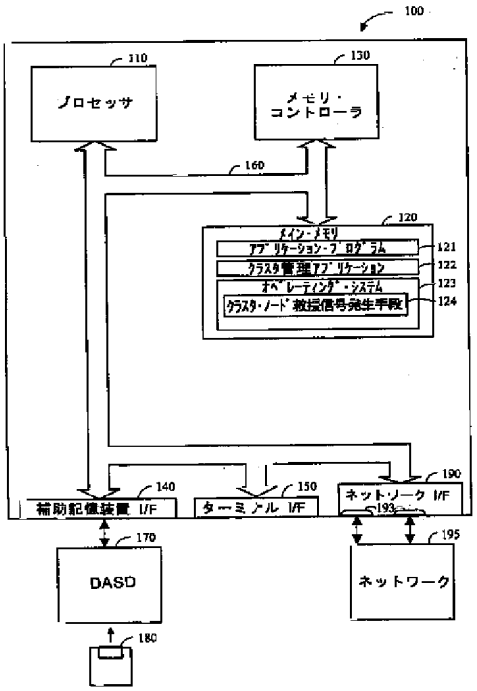
(31)優先権主張番号　０９／２８１０２６
(32)優先日　平成11年３月30日(1999.3.30)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレ－ション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）
(74)代理人　100086243
弁理士　坂口　博　（外１名）

最終頁に続く

(54)【発明の名称】　クラスタ・ノード救援信号発生システム

(57)【要約】　（修正有）
【課題】クラスタの信頼性を改良するクラスタ・ノード救援システム及び方法。
【解決手段】クラスタにおけるノードの障害発生時、クラスタ・ノード救援信号を供給する。これは、非コミュニケーティング・ノードが障害を生じたか、クラスタから仕切られただけなのかを、適切に決定することを可能にする。クラスタ・ノード救援システムは、ノードの差し迫った障害の検出時、クラスタにおける他のノードに迅速に送られるノード救援信号を供給し、そのノードが全面的に障害を生じる前にノード救援信号が発生する確率を改良する。ノード救援信号が効果的にクラスタに送られる時、クラスタは、そのノードが障害を生じたのであってクラスタから仕切られたのではないということを決定することができる。これは、管理者による少ない介入しか必要とせずに、クラスタが正しく一次的責任を他のノードに割り当てて応答することを可能にする。



【特許請求の範囲】
【請求項１】少なくとも１つのプロセッサと、
少なくとも１つのプロセッサに接続されたメモリと、
前記メモリ内にあって、クラスタにおけるノードの障害を表す事前形成された救援メッセージを含み、障害事象が検出された時に前記クラスタにおける他のノードに前記事前形成された救援メッセージを送るクラスタ・ノード救援信号発生手段と、
を含む装置。
【請求項２】前記クラスタ・ノード救援信号発生手段は、障害事象が検出された時、前記事前形成された救援メッセージを送るために待機する専用の救援信号実行タスクを含む、請求項１に記載の装置。
【請求項３】前記クラスタ・ノード救援信号発生手段は前記事前形成された救援メッセージを非同期的に送るためのメソッド及び前記事前形成された救援メッセージを同期的に送るためのメソッドを含む、請求項１に記載の装置。
【請求項４】前記事前形成された救援メッセージを非同期的に送るためのメソッドは現在の実行タスクを使用して前記事前形成された救援メッセージを送り、
前記事前形成された救援メッセージを同期的に送るためのメソッドは待機する専用の救援信号実行タスクを使用する、
請求項３に記載の装置。
【請求項５】前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時にはより順序正しいシャットダウン・プロシージャを可能にするために、前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用し、前記障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、請求項４に記載の装置。
【請求項６】前記クラスタ・ノード救援信号発生手段は前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するための機構を含む、請求項１に記載の装置。
【請求項７】前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの統合部分を含む、請求項１に記載の装置。
【請求項８】前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、請求項１に記載の装置。
【請求項９】少なくとも１つのプロセッサと、
少なくとも１つのプロセッサに接続されたメモリと、
前記メモリ内にあって、クラスタ・ノード装置の障害を表す事前形成された救援メッセージを含むクラスタ・ノード救援信号発生手段と、
を含み、
前記クラスタ・ノード救援信号発生手段は、現在の実行タスクを使用して前記事前形成された救援メッセージを非同期的に送るためのメソッド及び待機する専用の救援信号実行タスクを使用して前記事前形成された救援メッセージを同期的に送るためのメソッドを含み、
前記クラスタ・ノード救援信号発生手段は、障害事象が検出された時に前記クラスタにおける他のノードに前記事前形成された救援メッセージを送る、
クラスタ・ノード装置。
【請求項１０】前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時にはより順序正しいシャットダウン・プロシージャを可能にするために前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用し、前記障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、請求項９に記載のクラスタ・ノード装置。
【請求項１１】前記クラスタ・ノード救援信号発生手段は、前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記クラスタ・ノード装置がそれ自身を前記クラスタから除去することを保証するための機構を含む、請求項９に記載のクラスタ・ノード装置。
【請求項１２】前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの一部分である、請求項９に記載のクラスタ・ノード装置。
【請求項１３】前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、請求項９に記載のクラスタ・ノード装置。
【請求項１４】クラスタにおけるノードが障害事象を経験しようとしていることを表す事前形成された救援メッセージを提供するステップと、
差し迫った障害事象が検出された時、前記事前形成された救援メッセージを送るステップと、
を含む方法。
【請求項１５】前記事前形成された救援メッセージを処理して送るために専用の救援信号タスクを提供するステップを含む、請求項１４に記載の方法。
【請求項１６】前記事前形成された救援メッセージを送るステップは、前記差し迫った障害事象が存在する前に十分な時間がある時にはより順序正しいシャットダウン・プロシージャを可能にするために前記事前形成された救援メッセージを非同期的に送るステップ、及び前記差し迫った障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを同期的に送るステップを含む、請求項１４に記載の方法。
【請求項１７】前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するステップを含む、請求項１４に記載の方法。
【請求項１８】差し迫った障害事象のタイプを決定する

ステップを含む、請求項１４に記載の方法。
【請求項１９】クラスタにおけるノードが障害事象を経験しようとしていることを表す事前形成された救援メッセージを供給するステップと、
前記事前形成された救援メッセージを処理して送るために専用の救援信号タスクを供給するステップと、
差し迫った障害事象を検出するステップと、
前記検出された差し迫った障害事象が存在する前に十分な時間がある時には前記事前形成された救援メッセージを前記クラスタにおける他のノードに非同期的に送り、
前記検出された差し迫った障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを前記他のノードに同期的に送るステップと、
前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するステップと、
を含む方法。
【請求項２０】前記事前形成された救援メッセージを非同期的に送るステップは前記専用の救援信号タスクを使用し、
前記事前形成されたメッセージを同期的に送るステップは現在の実行タスクを使用する、
請求項１９に記載の方法。
【請求項２１】前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、請求項１９に記載の方法。
【請求項２２】（Ａ）クラスタにおけるノードの障害を表す事前形成された救援メッセージを含み、障害事象が検出された時に前記クラスタにおける他のノードに前記事前形成された救援メッセージを送るクラスタ・ノード救援信号発生手段と、
（Ｂ）前記クラスタ・ノード救援信号発生手段を保持する信号保持媒体と、
を含むプログラム製品。
【請求項２３】前記信号保持媒体は伝送媒体を含む、請求項２２に記載のプログラム製品。
【請求項２４】前記信号保持媒体は記録可能な媒体を含む、請求項２２に記載のプログラム製品。
【請求項２５】前記クラスタ・ノード救援信号発生手段は、障害事象が検出された時に前記事前形成された救援メッセージを送るための待機する専用の救援信号実行タスクを含む、請求項２２に記載のプログラム製品。
【請求項２６】前記クラスタ・ノード救援信号発生手段は前記事前形成された救援メッセージを非同期的に送るためのメソッド及び前記事前形成された救援信号を同期的に送るためのメソッドを含む、請求項２２に記載のプログラム製品。
【請求項２７】前記事前形成された救援メッセージを非同期的に送るためのメソッドは現在の実行タスクを使用して前記事前形成された救援メッセージを送り、
前記事前形成された救援メッセージを同期的に送るためのメソッドは待機する専用の救援信号実行タスクを使用する、
請求項２６に記載のプログラム製品。
【請求項２８】前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時には前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用してより順序正しいシャットダウン・プロシージャを可能にし、前記障害事象が存在する前に十分な時間がない時、前記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、請求項２７に記載のプログラム製品。
【請求項２９】前記クラスタ・ノード救援信号発生手段は、前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するための機構を含む、請求項２２に記載のプログラム製品。
【請求項３０】前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの統合部分を含む、請求項２２に記載のプログラム製品。
【請求項３１】前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、請求項２２に記載のプログラム製品。
【請求項３２】（Ａ）クラスタ・ノード装置の障害を表す事前形成された救援メッセージを含み、現在の実行タスクを使用して前記事前形成された救援メッセージを非同期的に送るためのメソッド及び待機する専用の救援信号実行タスクを使用して前記事前形成された救援メッセージを同期的に送るためのメソッドを含み、障害事象が検出された時、前記事前形成された救援メッセージを前記クラスタにおける他のノードに送るクラスタ・ノード救援信号発生手段、及び
（Ｂ）前記クラスタ・ノード救援信号発生手段を保持する信号保持媒体と、
を含むプログラム製品。
【請求項３３】前記信号保持媒体は伝送媒体を含む、請求項３２に記載のプログラム製品。
【請求項３４】前記信号保持媒体は記録可能な媒体を含む、請求項３２に記載のプログラム製品。
【請求項３５】前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時には前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用してより順序正しいシャットダウン・プロシージャを可能にし、前記障害事象が存在する前に十分な時間がない時には記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、請求項３２に記載のプログラム製品。
【請求項３６】前記クラスタ・ノード救援信号発生手段は、前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記クラスタ・ノード装置がそれ

自身を前記クラスタから除去することを保証するための機構を含む、請求項３２に記載のプログラム製品。
【請求項３７】前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの統合部分を含む、請求項３２に記載のプログラム製品。
【請求項３８】前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、請求項３２に記載のプログラム製品。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、概していえば、コンピュータをクラスタ化することに関し、詳しくいえば、クラスタ通信のための救援信号発生に関するものである。
【０００２】
【従来の技術】この電子時代における多くのタイプの情報のために、社会はコンピュータ・システムに依存している。ハードウエア（例えば、半導体、回路ボード等）及びソフトウエア（例えば、コンピュータ・プログラム）の種々の組み合わせに基づいて、コンピュータ・システムは設計が広範囲に変わる。今日の多くのコンピュータ・システムは、他のコンピュータ・システムと「ネットワーク化する」ように設計されている。ネットワークキングを通して、単一のコンピュータ・システムが他のコンピュータ・システムにおいて記憶及び処理された情報をアクセスすることができる。従って、ネットワーキングの結果、極めて多数のコンピュータ・システムが極めて多数の電子リソースへのアクセスを持つことになる。
【０００３】ネットワーキングは、コンピュータ・システム相互間の物理的「ルート」及び通信「プロトコル」に関する合意を得た使用によって可能にされる。どのようなプロトコルが選択されるかは、ネットワーク化されたコンピュータ・システムの数、それらのコンピュータ・システムを隔てている距離、及びそれらのコンピュータシステム相互間における情報交換という目的を含む要因に依存する。少数のコンピュータ・システムだけが近接してネットワーク化されている場合、通信プロトコルは極めて単純なものになり得る。しかし、非常に多数のコンピュータ・システムが付加されている時、及びコンピュータ・システムが大きな距離よって隔てられている時、これらの通信プロトコルは更に複雑なものになる。
【０００４】通信プロトコルの複雑さは情報交換のタイプによっても変わる。例えば、或るプロトコルは大量の情報を送る時の精度を高めるし、一方、別のプロトコルは情報転送の速度を強化する。コンピュータ・システム・ネットワーク上で稼働するアプリケーションの通信要件は、どのようなタイプのプロトコルが選択されるかを決定する。リアル・タイムの、しかも信頼性の高い情報転送を必要とするコンピュータ・アプリケーションの１つの例は「クラスタ」管理アプリケーションである。
【０００５】クラスタ化は、リソースの連続した使用可能性を提供するための及びワークロードを分配するためのコンピュータ・システムのネットワーキングである。コンピュータ・システムのクラスタは、コンピュータ・システム・ユーザの視点からは１つのコンピュータ・システムのように見えるが、実際には、相互にバックアップするコンピュータ・システムのネットワークである。クラスタ内の１つのコンピュータ・システムにおけるオーバロード又は障害の場合、クラスタ管理アプリケーションは、障害のあるコンピュータ・システムに対する処理責任を、そのクラスタにおける他のコンピュータ・システムに自動的に再割り振りする。従って、ユーザの視点からは、リソースの使用可能性における割り込みは存在しない。
【０００６】一般には、クラスタにおける１つのノードがアプリケーション（例えば、データベース、サーバ）に対する一次的責任を割り振られ、他のノードがバックアップ責任を割り振られる。或るアプリケーションに対する一次ノードが障害を生じた時、そのクラスタにおけるバックアップ・ノードがそのアプリケーションに対する責任を引き継ぐ。これは、そのアプリケーションの高い使用可能性を保証する。
【０００７】クラスタ化は、クラスタ管理アプリケーション・プログラムが１つのクラスタにおける各コンピュータ・システム上で稼働することによって可能にされる。これらのアプリケーションは、クラスタのアクティビティを制御するためにクラスタ・ネットワーク全域にわたってクラス・メッセージの往来を中継する。クラスタ・メッセージングは、そのクラスタにおけるどのコンピュータシステムがどのような一次的責任及びバックアップ責任を有するかに関する最新情報を分配するためにも使用される。
【０００８】クラスタにおいて作動するアプリケーションの高い使用可能性を保証するために、そのクラスタは、クラスタにおけるすべてのノードのステータスを追跡できることが必要である。これを行うために、クラスタにおける各コンピュータ・システムは、同じクラスタにおける他の各コンピュータ・システムを連続的に監視し、各々が活性状態であること及びそれに割り当てられた処理を遂行しようとしていることを保証する。従って、クラスタにおける或るノードが障害を生じた時、それの一次的責任をバックアップ・ノードに割り当てることが可能である。
【０００９】残念ながら、クラスにおける或るノードが障害を生じたことを知らせることがいつも可能であるわけでない。例えば、或るノードとそのクラスタにおける残りのノードとの間のネットワーク接続が障害を生じた場合、クラスタは、そのノードが適正に動作しているかどうかを知らせることが最早できないであろう。或るノ

ードが依然として動作しているがそのクラスタにおける他のノードへのそれのネットワーク接続が障害を生じた場合、そのノードはクラスタから仕切られているといえる。或るノードがそのクラスタにおける残りのノードとコミュニケートすることを不意に停止した場合、そのノードが障害を生じたのか或いは単にそのクラスにおける残りのノードから仕切られただけなのかを容易に決定することはできない。そのノードが障害を生じたものとクラスタが誤って仮定し、しかもアプリケーションに対する一次的責任をバックアップ・ノードに割り当てる場合、クラスタは、２つのノードの両方とも、それらが一次ノードであるものと信じさせたままにすることがあり得る。この結果、両方のノードがクラスタに対するリクエストに応答するので、データベースにおけるデータ不整合が生じることがあり得る。一方、そのノードが依然としてそれの一次アプリケーションを遂行しており、しかも単にクラスタから仕切られただけであって、バックアップ・ノードに一次的責任を割り当ててないものとクラスタが誤って仮定した場合、それらのアプリケーションは最早そのクラスタのクライアントにとって利用可能ではないであろう。従って、多くの場合、クラスタは、管理者による手操作介入なくして非コミュニケーティング・ノードに正しく応答することができない。

【００１０】

【発明が解決しようとする課題】より多くのリソースがコンピュータ・システム・ネットワーク全域にわたってアクセス可能になる時、そのようなネットワーク・リソースへの連続的なアクセスに対する要求が増えるであろう。そのようなネットワーク・リソースに連続的な使用可能性を与えるための手段として、クラスタに対する要求も対応して増えるであろう。クラスタのノードのステータスを決定する改良された方法がない場合、これらのリソースに対する連続的な使用可能性は十分には実現されないであろう。

【００１１】

【課題を解決するための手段】本発明によれば、クラスタの信頼性を改良するクラスタ・ノード救援システムが提供される。そのクラスタ・ノード救援システムは、クラスタにおけるノードが障害を生じようとしている時、クラスタ・ノード救援信号を供給する。これは、非コミュニケーティング・ノードが障害を生じたか或いは単にクラスタから仕切られただけなのかどうかを、そのクラスタがより適切に決定することを可能にする。望ましいクラスタ・ノード救援システムがオペレーティング・システムに深く組み込まれ、そのノードの差し迫った障害が検出される時、クラスタにおける他のノードに迅速に送られる事前形成されたノード救援信号を供給する。これは、そのノードが全面的に障害を生じる前にノード救援信号が発生する確率を改良する。ノード救援信号が効果的にクラスタに送られる時、クラスタは、そのノードが障害を生じたのであってクラスタから仕切られたのではないということを正確に決定することができる。これは、クラスタが正しく、即ち、一次的責任を他のノードに割り当てることによって応答することを可能にし、しかも管理者による少ない介入しか必要としない。従って、望ましい実施例は、クラスタの信頼性の改良及び管理者に対する依存度の減少を提供する。

【００１２】本発明の上記及び他の特徴及び利点は、本発明の望ましい実施例において述べられるように、しかも添付図面に示されるように、下記の更に詳細な説明から明らかであろう。

【００１３】

【発明の実施の形態】本発明はクラスタ通信に関するものである。クラスタ化の概念に一般的に精通してない個人のために、下記の「概要」セクションは、本発明の望ましい実施例を理解する助けになると思われる多くの基本的な概念及び用語を示す。クラスタ化の分野における当業者は、この「概要」をスキップして本明細書の「詳細な説明」のセクションに直接に進んでもよい。

【００１４】１．概要

クラスタ化は、コンピュータが作業を分担すること及び相互にバックアップとして作用することを可能にするという意味でコンピュータ又はコンピュータのグループを連係させることである。このように、クラスタは、たとえクラスタにおける１つ又は複数のコンピュータが障害を生じてもコンピュータ・システムが動作し続けること及びサービスを提供し続けることを可能にする。コンピュータ・ユーザの視点から、コンピュータ・システムのクラスタは１つのコンピュータ・システムのように見える。クラスタ化はコンピュータ・クラスタのユーザにとって透明であり、それらのユーザは、それらが１つのコンピュータ・システムを使用しているのか又は複数のコンピュータ・システムを使用しているのかに関して知る必要がない。その代わり、コンピュータ・クラスタのユーザにとって重要なことは、そのようなデータベース、プリンタ、ファイル等のようなそれらが必要とするリソースへのアクセスをそれらが有するということである。コンピュータ・システムをクラスタ化することによって、必要なリソースに対する連続的な使用可能性を得ることが可能になる。

【００１５】コンピュータ・システムをクラスタ化することには多くの利点がある。第１に、しかも最も重要なこととして、クラスタは、クラスタ内のコンピュータ・システムが相互にバックアップすることを可能にすることによって、より高い使用可能性を提供する。第２に、クラスタ化は、処理パワーを改良するために必要とされるだけの付加的なコンピュータ・システムが追加されることを可能にすることによって、拡張容易性（ｓｃａｌａｂｉｌｉｔｙ）を増大させる。第３に、クラスタにおけるコンピュータ・システム相互間でワークロードを平

衡させることが可能である。

【００１６】クラスタを形成するコンピュータ・システムは「ノード」とも呼ばれる。一般に、ノードという用語は、プロセッサ、通信コントローラ、又はターミナルを呼ぶことがある。しかし、クラスタを目的とする場合、ノードはクラスタにおける個々のコンピュータ・システムの１つを呼ぶ。一般に、クラスタにおける各ノードは、そのクラスタの支持で一次的責任及びバックアップ責任を割り当てられる。割り当てられる責任は、データへのアクセスを行うこと、コンピュータ・アプリケーションを実行すること、或いは、プリンタ、スキャナ、又はファックス・マシンのようなハードウエア・リソースへのアクセスを行うことのような１つ又は複数の機能に対するものであってもよい。クラスタにおけるノードは、すべてのノードが機能しているということ、即ち、各ノードにおけるクラスタリング・ソフトウエアが活動的であること及び一次的責任からバックアップ責任への切り替えを必要とする条件を積極的に監視することを保証するためにコミュニケートする。

【００１７】クラスタにおけるノードは一次的責任及びバックアップ責任を割り当てられる。各アプリケーションに対する一次ノードは、タスクを遂行し且つそのクラスタのクライアントと対話するノードである。一次ノードがそれの割り当てられた機能を遂行することができなくなる時、クラスタ管理アプリケーションは、遂行することができないノードに割り当てられたリソースへのアクセスをそのクラスタが依然として有することを保証するように作用しなければならない。これは、そのリソースに対するバックアップ・ノードの１つを一次的責任に切り替えることを伴う。この場合、クラスタ・ユーザは、たとえ必要なリソースを提供する一次的責任を持つコンピュータ・システムが利用可能でない時でも、依然として、その必要なリソースへのアクセスを有する。

【００１８】クラスタ管理アプリケーション及びすべてのノード間の通信設備は、クラスタがユーザの視点からは単一のコンピュータ・システムとして動作することを可能にする。例えば、メッセージは、クラスタにおける他のノードに関する状況を知らせるためにすべてのノードに送られる。メッセージは、どのノードが特定のアプリケーションに対する一次的責任及びバックアップ責任を有するかに関して最新のものを追跡するためにすべてのノードに送られる。これは、複数のノードが特定のアプリケーションに対する一次ノードとして振る舞おうとするというように、複数のノードが矛盾したオペレーションを遂行することのないようにする。２つのノードが共に一次ノードであると考えて動作することが許される場合、データの不整合のような問題が生じることがある。従って、１つのノードがそれの割り当てられた責任を遂行することができない時にどのようなアクションを取るべきかに関してすべてのノードが同意状態にあるように、メッセージがすべてのノードに送られる。クラスタが適切に機能することを保証するために、クラスタにおけるすべてのノードがこれらのクラスタ・メッセージを正しい順序で受けなければならない。

【００１９】基本的なクラスタ・メッセージの１つのタイプは、「ハートビート」と呼ばれる。ハートビートは、クラスタにおけるノード相互間で送られる低レベルのメッセージであり、どのノードが現在適正に遂行しているかをクラスタが追跡することを可能にする。例えば、各ノードは、一般には、ハートビート信号を論理的に隣接したノードに規則的な間隔で送るであろう。従って、クラスタにおける各ノードは、それの論理的に隣接したノードからハートビート信号をこれらの同じ規則的な間隔で受けるものと期待する。ノードが適正なハートビート信号を或る延長した期間の間に受けなかった場合、そのノードは、それの近隣のノードに関して潜在的な問題が存在することを知る。このようにハートビートを受けることができないことが継続する場合、クラスタ管理システムは適切なアクションを取ろうとするであろう。

【００２０】クラスタがそれのノードを監視するもう１つの方法はメッセージ・タイマによるものである。例示的なクラスタ化・システムでは、或るノードに送られたメッセージが障害を生じた場合、それは、設定された期間の間自動的に再試行されるであろう。更に、複数の試行の後に、メッセージが依然として配送されない場合、クラスタ管理システムは、再び問題があることを知るであろうし、適切なアクションを取ろうとするであろう。

【００２１】残念ながら、取るべき適切なアクションがどのようなものであるかをクラスタ管理システムが知るということがいつも可能であるわけではない。例えば、ノードとクラスタにおける残りのノードとの間のネットワーク接続が障害を生じた場合、クラスタは、そのノードが適正に動作しているかどうかを最早知らせることができないであろう。ノードが依然として動作しているが、クラスタにおける他のノードへのそれのネットワーク接続が障害を生じた場合、そのノードはクラスタから仕切られているといえる。或るノードがクラスにおける残りのノードとコミュニケートすることを不意に停止する時、そのノードが障害を生じたのか、或いは単にクラスタにおける残りのノードから仕切られただけであるのかを容易に決定することはできない。ノードが単に仕切られただけである時にそのノードが障害を生じたものとクラスタが誤って仮定し、その仕切られたノードのアプリケーションに対する一次的責任をバックアップ・ノードに割り当てる場合、クラスタは、２つのノードの両方とも、それらが一次ノードであるものと信じさせたままにすることがあり得る。更に、両方のノードはクラスタへのリクエストに応答するので、これはデータの不整合を生じさせることがある。

【００２２】一方、そのノードが実際に障害を生じた時にそれが仕切られていたものとクラスタが誤って仮定した場合、それらのアプリケーションは最早そのクラスタのクライアントには利用可能ではないであろう。従って、多くの場合、クラスタは、管理者による手操作介入なくして非コミュニケーティング・ノードに正しく応答することができない。

【００２３】２．詳細な説明

本発明によれば、クラスタの信頼性を改良するクラスタ・ノード救援システムが提供される。クラスタ・ノード救援システムは、クラスタにおけるノードが障害を生じようとしている時、クラスタ・ノード救援信号を供給する。これは、非コミュニケーティング・ノードが障害を生じたのか又は単にクラスタから仕切られただけであるのかをクラスタがより適切に決定することを可能にする。望ましいクラスタ・ノード救援システムはオペレーティング・システムに深く組み込まれ、そのノードの差し迫った障害が検出された時、それはクラスタにおける他のノードに迅速に送出可能な事前形成されたノード救援信号を発生する。これは、ノードが全面的に障害を生じる前にノード救援信号が発生する確率を改善する。ノード救援信号がクラスタに有効に送られる時、クラスタは、ノードが障害を生じ、しかもそのクラスタから仕切られてないということを正確に決定することができる。これは、クラスタが正しく、即ち、他のノードに一次的責任を割り当てることにより、管理者のより少ない介入でもって応答することを可能にする。従って、望ましい実施例はクラスタの信頼性の改善及び管理者への依存度の減少をもたらす。

【００２４】図１を参照すると、本発明の望ましい実施例によるコンピュータ・システム１００はＡＳ／４００中型コンピュータ・システムである。しかし、本発明の方法及び装置が任意にコンピュータ・システムに、そのコンピュータ・システムが複雑なマルチユーザ・コンピューティング装置であるか、或いは、パーソナル・コンピュータ又はワークステーションのようなシングル・ユーザの装置であるかに関係なく、等しく適用することは当業者には明らかであろう。例えば、これらの機能がＩＢＭ社のＯＳ／２、ＯＳ／３９０、及びＲＳ／６０００、マイクロソフト社のＷｉｎｄｏｗｓＮＴ、ノベル社のＮｅｔＷａｒｅ、Ｌｉｎｕｘ、及び他の種々のＵｎｉｘの変種のような他のシステムに設けられてもよいことは当業者には明らかであろう。コンピュータ・システム１００は、プロセッサ１１０、メイン・メモリ１２０、メモリ・コントローラ１３０、補助記憶装置インターフェース１４０、ターミナル・インターフェース１５０、及びネットワーク・インターフェース１９０を適宜に含む。これらの装置はすべてシステム・バス１６０を介して相互接続される。図１に示されたコンピュータ・システム１００に対して、キャッシュ・メモリ又は他の周辺装置の追加のような種々の修正、追加、又は削除を、本発明の技術的範囲において行い得ることに留意してほしい。図１は、コンピュータ・システム１００の顕著な特徴のいくつかを簡単に説明するために示される。

【００２５】プロセッサ１１０はコンピュータ・システム１００の計算機能及び制御機能を遂行し、適当な中央処理ユニット（ＣＰＵ）を含む。プロセッサ１１０は、マイクロプロセッサのような単一の集積回路を含み得るし、或いはプロセッサの機能を達成するために協同して働く任意の適当な数の集積回路又は回路ボードを含み得るものである。プロセッサ１１０は、必要に応じてメイン・メモリ１２０におけるコンピュータ・プログラムを適切に実行する。

【００２６】補助記憶装置インターフェース１４０は、コンピュータ・システム１００が磁気ディスク（例えば、ハード・ディスク又はフロッピ・ディスク）又は光学的記憶装置（例えば、ＣＤ－ＲＯＭ）のような補助記憶装置に情報を記憶し及び補助記憶装置から情報を検索することを可能にする。１つの適切な記憶装置はダイレクト・アクセス記憶装置（ＤＡＳＤ）１７０である。図１に示されるように、ＤＡＳＤ１７０は、フロッピ・ディスク１８０からプログラム及びデータを読み取ることができるフロッピ・ディスク・ドライブであってもよい。完全に機能的なコンピュータ・システムに関連して本発明を説明した（及び説明を続ける）けれども、本発明の機構がプログラム製品として種々な形式で配布可能であること、及び本発明が、実際に配布を行うためには特定のタイプの信号保持媒体に関係なく、等しく適用することは当業者には明らかであろう。信号保持媒体の例は、フロッピ・ディスク（例えば、ディスク１８０）及びＣＤ－ＲＯＭのような記録可能なタイプの媒体及びワイヤレス通信リンクを含むディジタル及びアナログ通信・リンクのような伝送タイプの媒体を含む。

【００２７】メモリ・コントローラ１３０は、プロセッサ１１０とは別のプロセッサ（図示されてない）を使用することによって、リクエストされた情報をメイン・メモリ１２０から又は補助記憶装置インターフェース１４０を通してプロセッサ１１０に移動させる責任がある。説明の便宜上、メモリ・コントローラ１３０は別個のエンティティとして示されるけれども、実際には、メモリ・コントローラ１３０によって与えられる機能の一部分がプロセッサ１１０、メイン・メモリ１２０、又は補助記憶インターフェース１４０と関連する回路にあってもよいことは当業者には明らかであろう。

【００２８】ターミナル・インターフェース１５０は、システム管理者及びコンピュータ・プログラマが、通常はプログラム可能なワークステーションを介してコンピュータ・システム１００とコミュニケートすることを可能にする。図１に示されたシステム１００は単一のメイン・プロセッサ１１０及び単一のシステム・バス１６０

しか含まないけれども、複数のプロセッサ及び複数のシステム・バスを有するコンピュータ・システムにも本発明が同様に適用することを理解すべきである。同様に、その望ましい実施例のシステム・バス１６０は一般的に配線されたマルチドロップ・バスであるけれども、コンピュータ関連の環境における双方向通信をサポートする任意の接続手段が使用されてもよい。

【００２９】ネットワーク・インターフェース１９０はコンピュータ・システム１００とネットワーク１９５におけるリモート・コンピュータ・システムとの間の情報の転送をサポートする。望ましい実施例では、ネットワーク１９５における１つ又は複数のノードが、クラスタとしてコンピュータ・システム１００と共に働くように同様に設定される。ネットワーク１９０は1つ又は複数のネットワーク・インターフェース・アダプタ１９３を適宜に含み、ネットワーク・インターフェース・アダプタ１９３の各々は、一般に、コンピュータ・システム１００のようなコンピュータ・システムに容易に付加することができる拡張カードとして実装される。ネットワーク・インターフェース・アダプタ１９３の例は、ペリフェラル・コンポーネント・インターコネクト（ＰＣＩ）拡張カード、インダストリ・スタンダード・アーキテクチャ（ＩＳＡ）拡張カード、財産権のあるアダプタ・カード、及び現在知られている又は将来発明される任意のタイプのアダプタを含む。ネットワーク・インターフェース１９０の機能がメイン・メモリ１２０及びプロセッサ１１０の一部として直接にインプリメント可能であることは当業者には明らかであろう。ネットワーク１９５は当業者に知られた任意のタイプのネットワークを表す。これは、インターネット、イントラネット、ローカル・エリア・ネットワーク（ＬＡＮ）、広域ネットワーク（ＷＡＮ）、又はコンピュータ・システムを相互にコミュニケートさせるための現在知られている又は将来開発される任意の構成のハードウエア及びソフトウエアを含む。ネットワーク１９５上には、クラスタにおける他のノードが存在するであろう。

【００３０】メイン・メモリ１２０は１つ又は複数のアプリケーション・プログラム１２１、クラスタ管理アプリケーション１２２、及びクラスタ・ノード救援信号発生手段１２４を含むオペレーティング・システム１２３を適宜に含む。メモリ１２０におけるこれらのプログラムはすべてそれの最も広い意味において使用され、ソース・コード、中間コード、マシン・コード、及び他の任意の表示のコンピュータ・プログラムを含む任意の及びすべての形式のコンピュータ・プログラムを含む。

【００３１】その望ましい実施例では、アプリケーション・プログラム１２１は、信頼性及び拡張容易性の向上を提供するためにクラスタ化を使用する任意のプログラムを含み得るものである。このように、アプリケーション・プログラム１２１は、一般に、コンピュータ・システム１００を一次ノード又はバックアップ・ノードとするすべてのプログラムを含むであろう。そのようなアプリケーション・プログラムの例は、ウェブ・サーバ、ファイル・サーバ、データベース・サーバ等を含む。

【００３２】クラスタ管理アプリケーション１２２は、クラスタを作成及び管理するために必要な機構を提供する。これは、コンピュータ・クラスタの管理を求める管理的なリクエストの処理を含むであろう。例えば、これは、クラスタを作成するための機構、クラスタにノードを付加するための機構、及びクラスタからノードを除去するための機構等を含むことが望ましい。

【００３３】その望ましい実施例では、クラスタ・ノード救援信号手段１２４はオペレーティング・システム１２３と統合され、或るノードの差し迫った障害が検出された時にノード救援信号を送出するための最も速い且つ最も効率的な手段を提供する。

【００３４】メイン・メモリ１２０がいつも示されたすべての機構のすべての部分を必ずしも含むわけではないことは理解されるべきである。例えば、アプリケーション・プログラム１２１、クラスタ管理アプリケーション１２２、及びオペレーティング・システム１２３の一部分は、プロセッサ１１０が実行するために命令キャッシュ（図示されてない）にロードされてもよく、一方、別のファイルが磁気ディスク記憶装置又は光ディスク記憶装置（図示されてない）に完全に記憶されてもよい。更に、コンピュータ・プログラムはすべて同じメモリ・ロケーションにあるように示されているけれども、メイン・メモリ１２０が異種のメモリ・ロケーションより成るものでもよいことは勿論である。本願において使用される「メモリ」という用語は、システム１００の仮想メモリ空間における任意の記憶ロケーションを呼ぶ。

【００３５】コンピュータ・システム１００がクラスタにおける各ノードの例示的なものであること、及びそのクラスタにおける各ノードが、それの障害の場合にそのクラスタにおける他のノードにノード救援信号を迅速に送る能力を有することも理解されるべきである。そこで、他の各ノードにおけるクラスタ管理アプリケーション１２２が、そのクラスタにおける他のノードに適切な一次的責任を割り当てることによって適切に応答することが可能である。

【００３６】次に、図２を参照すると、クラスタ・ノード救援信号発生手段１２４の望ましい実施例が更に詳細に示される。上述のように、クラスタ・ノード救援信号発生手段１２４は、ノードの差し迫った障害時にクラスタ内の他のノードに救援信号を送ることを必要とするメカニズムを提供する。これは、クラスタ管理アプリケーション１２２（そのクラスタの他のノードにおける）が非応答ノードが障害を生じ且つクラスタからまだ仕切られてないことを正確に決定する。

【００３７】望ましい実施例では、クラスタ・ノード救

援信号発生手段１２４はオペレーティング・システム１２３に統合され、それがノードの差し迫った障害に迅速に応答することを可能にする。望ましい実施例では、クラスタ・ノード救援信号発生手段１２４は救援信号送出メソッド、事前形成された救援メッセージ、及びそのメッセージを送るための専用の救援信号を含む。

【００３８】最も望ましい実施例では、救援信号送出メソッドの２つの利用可能な実施方法がある。１つのメソッドは救援メッセージを同期的に送ることであり、それは、そのメッセージが送られたということが確認されるまで、シャットダウン中に遂行されるべき他のすべての方法が待機状態にされることを意味する。もう１つのメソッドは救援信号を非同期的に送ることであり、それは、クラスタ・ノードが救援信号送出メソッドの開始後に他のタスクを処理し続けることができるということである。

【００３９】望ましい実施例では、その非同期的メソッドは、ノードを順序正しくシャットダウンするに十分な時間がある時に使用される。その非同期的メソッドを使用することは、救援メッセージが送られつつある間、現在の実行スレッドがシャットダウンに備えて他のタスクを遂行することを可能にする。救援信号を作成及び送出する間そのノードの現在の実行スレッドが他のタスクを遂行し続けることを可能にすることは、その結果として、より順序正しいシャットダウンを生じるという利点を有するが、救援信号が実際に送られる前に潜在的に長い遅延を生じるという欠点を有する。逆に、同期的メソッドは、障害が差し迫っていて、救援メッセージが直ちに送られなければならない時に望ましい。その同期的メソッドは、救援メッセージが送られるまで、現在の実行スレッドに関する他のすべての処理を待機状態におき、その結果としてノード救援メッセージの迅速な送出を生じさせる。

【００４０】例えば、障害事象が停電であって、バッテリのバックアップ電力がある場合、救援メッセージが非同期的に送られることを可能にするための比較的多くの時間が障害前に存在するであろう。これは、他のシャットダウンプロシージャのようなより多くの並列的アクションが遂行されて、より順序正しいシャットダウンを提供することを可能にする。

【００４１】もう１つの例として、障害事象がハードウエア障害であるか、又はＩＰスタックの終了である場合、待つべき時間がないかもしれず、救援メッセージを同期的に送ることが望ましい。これは、事前形成された救援メッセージの即時送出を生じさせて、そのメッセージが送られるまで現在の実行スレッドが他のプロセスに進むことがないようにする。これは、救援メッセージができるだけ速く送られることを可能にし、場合によっては、救援メッセージが送られる前にノードがシャットダウンしないようにする。

【００４２】その望ましい実施例では、同期的救援信号メッセージが現在の実行タスクにおいて処理され、メッセージが更に迅速に送られることを可能にする。更に詳しく云えば、現在のタスクはそのメッセージを直ちに送ることができ、一方、待機する専用の救援タスクは、そのメッセージが送られる前に呼び出されなければならないであろう。しかし、現在のタスクは、救援メッセージが送られるまでそれが進行することを可能にすることなくそのメッセージを送るために使用されるので、遂行される必要のある他のアクションが待機状態になるであろう。従って、救援信号メッセージを送るために同期的メソッドを使用することは、障害事象が生じる前に、しかし他のプロセスを犠牲にして、メッセージが送られる機会を改善する。

【００４３】逆に、非同期的救援信号メッセージは専用の救援信号タスクのためのタスク・キューに送られ、それに関連して作動する。一般に、このタスクはそのメッセージの送出前に遂行するために呼び出される必要があるであろうが、それのキューには他になにも存在しないので、それは非専用タスクを使用するよりも依然として速いであろう。一旦、非同期呼び出しが専用の救援タスクに対して行われると、現在のタスクは、救援メッセージが送られるのを待つことなく、それの別のプロセスを続けることができる。

【００４４】そのメッセージを送るための専用の救援信号タスクは、オペレーティング・システムにおける１つのプロセスとして機能する低レベルの実行スレッドを含むことが望ましい。これは、呼び出された時に実行されるのを待つ１つのインスタンス化されたタスク・オブジェクトとしてインプリメント可能である。その非同期救援信号メソッドが呼び出される時、それはタスク・オブジェクト・メッセージ・キューを呼び出す。利用可能な次のプロセッサはこのメソッドによって定義されたコードをピックアップし、それを実行するであろう。そのノード救援信号を送るための専用のタスクが存在するので、そのメソッドがタスク・オブジェクト・メッセージ・キューにおいて待機しなければならない可能性はあり得ない。その代わり、それは次の利用可能なプロセッサによって実行されるであろう。一般的なオペレーティング・システムは、任意の所与の時間に実行するために利用可能な多くの種々なタスクを有するが、一時に１つのタスクしか各プロセッサによって実行され得ない。プロセッサが何かを待たなければならない時、現在のタスクは無視され、プロセッサは次のタスクに進む。

【００４５】事前形成された救援メッセージは、望ましくは、障害事象の場合に送られる準備ができている予めインスタンス化されたメッセージ・オブジェクトを含む。そのメッセージは、それがノード救援メッセージあることを表すヘッダ及びその救援メッセージを送ったノードのＩＤを含むことが望ましい。更に、そのメッセー

ジは、それの障害の理由がわかる場合にはそれに関するデータを、たとえこれが必要なくても、含むことが可能である。

【００４６】次に図３を参照すると、望ましい実施例に従ってノード救援信号を送るための方法５００が示される。第１ステップ５０２はクラスタ・ノードが障害事象を経験した時のステップである。次のステップ５０４は、そのノードのシステムが障害を検出し、クラスタ・ノード救援信号発生手段における救援信号メソッドを呼び出すステップである。

【００４７】望ましい実施例では、実際には、正しい応答に対するすべてのタイプの障害事象を所定時間内に十分に検出することはできないけれども、いくつかのタイプの障害事象を検出することはできる。例えば、１つのタイプの障害事象は停電である。ノード・コンピュータ・システムは、クラスタ・ノード救援信号を送るには十分に長いオペレーションを維持しながら停電を検出することができることが望ましい。上述のように、クラスタ・ノード救援信号は事前形成された信号を含むので、ノード救援信号は、従来の機構が可能であった速度よりもずっと速く送られる。これは、ノードが完全に障害を生じてしまう前にそのメッセージを送るための時間内に障害事象が検出される可能性を増大させる。障害事象のもう１つの例として、ノード・コンピュータ・システムが救援メッセージを呼び出し且つ送出するようになるずっと前に、そのノードの他のメンバとコミュニケートするために使用されるプロトコル・スタックの分解（ｔａｋｅｄｏｗｎ）をそのノード・システムが検出することができる。

【００４８】勿論、これらは、クラスタ・ノード救援信号を送るに十分な高度の警報を備えたノード・システムによって検出され得る３つのタイプの障害事象である。他の障害事象はオペレーティング・システムにおける切迫したクラッシュ、１つ又は複数のハードウエア・コンポーネント（例えば、ドライブ、ネットワーク・アダプタ等）であってもよい。従って、本発明の望ましい実施例が、救援信号を送るには十分に前に予め検出可能な任意のタイプの障害に適用可能であること、及び事前形成された救援信号を迅速に送出する望ましい実施例の能力が従来のシステムおいて必要とされた事前警報をかなり減少させるということは当業者には明らかであろう。

【００４９】望ましくは、オペレーティング・システムは、障害が生じる前の時間の大きさに従って、それが適切な救援信号送出メソッド（即ち、非同期的な又は同期的な）を呼び出すことができるように、発生しようとしている障害事象のタイプを認識する。更に、オペレーティング・システムがその障害のタイプを、非同期的に応答される障害のタイプであると認識しなかった場合、それは、障害が生じる前に救援メッセージが送られることを保証するために同期的メソッドを呼び出すことが望ましい。

【００５０】次のステップ５０６は、専用の救援信号タスクを使用して１つの事前形成された救援信号をクラスタにおける任意の視聴者に送るためのステップである。その事前形成された救援信号は、そのメッセージを送るノードの名前を含むインスタンス化されたメッセージ・オブジェクトを含むことが望ましい。これは、そのメッセージが最初に形成されるのを待つことなく、救援信号がプロトコル・スタック上に迅速に置かれることを可能にする。

【００５１】望ましい実施例では、ステップ５０６は、検出された障害事象のタイプ及び事前形成された救援信号を送るための残っている時間の量に従って、非同期的又は同期的に遂行可能である。

【００５２】次のステップ５０８は、ノードがすべての一次的責任及びバックアップ責任を中止することを救援信号メソッドが保証するためのステップである。更に、複数のノードすべてが或るアプリケーションに対する一次機能を遂行しようとすることは望ましくない。従って、ノード救援信号を送ることによって、このノードは、それが最早利用可能ではなく、適切なバックアップ・ノードによって置換されなければならないということを他のノードに知らせる。何らかの理由でこのノードが回復する場合、おそらく他のノードがそれの以前からの一次ノードを遂行し始めていると思われるので、それは依然としてそれ自身をオフラインであると見なさなければならない。

【００５３】従って、本発明の望ましい実施例は、クラスタ・ノード救援システム及びクラスタの信頼性を改善する方法を提供する。クラスタ・ノード救援システムは、クラスタ上の或るノードが障害を生じようとしている時、クラスタ・ノード救援信号を供給する。これは、非コミュニケーティング・ノードが障害を生じたか又は単にクラスタから仕切られただけであるかをクラスタがより的確に決定することを可能にする。その望ましいクラスタ・ノード救援システムはオペレーティング・システムに深く組み込まれ、そのノードの差し迫った障害が検出された時にクラスタにおける他のノードに迅速に送られる事前形成されたノード救援信号を供給し、そのノードが全面的に障害を生じる前にノード救援信号が送出される確率を改善する。ノード救援信号がクラスタに効果的に送られる時、クラスタは、そのノードが障害を生じてそのクラスタから未だ仕切られてないことを正確に決定することができる。これは、クラスタが正しく、即ち、他のノードに一次的責任を割り当てることによって応答することを可能にし、管理者による必要な介入を少なくする。

【００５４】本発明を、それの望ましい実施例に関して詳しく示し、説明したけれども、本発明の精神及び技術的範囲から逸脱することなく形式及び詳細に関する種々

の変更を行い得ることは当業者には明らかであろう。
【００５５】まとめとして、本発明の構成に関して以下の事項を開示する。
【００５６】（１）少なくとも１つのプロセッサと、少なくとも１つのプロセッサに接続されたメモリと、前記メモリ内にあって、クラスタにおけるノードの障害を表す事前形成された救援メッセージを含み、障害事象が検出された時に前記クラスタにおける他のノードに前記事前形成された救援メッセージを送るクラスタ・ノード救援信号発生手段と、を含む装置。
（２）前記クラスタ・ノード救援信号発生手段は、障害事象が検出された時、前記事前形成された救援メッセージを送るために待機する専用の救援信号実行タスクを含む、上記（１）に記載の装置。
（３）前記クラスタ・ノード救援信号発生手段は前記事前形成された救援メッセージを非同期的に送るためのメソッド及び前記事前形成された救援メッセージを同期的に送るためのメソッドを含む、上記（１）に記載の装置。
（４）前記事前形成された救援メッセージを非同期的に送るためのメソッドは現在の実行タスクを使用して前記事前形成された救援メッセージを送り、前記事前形成された救援メッセージを同期的に送るためのメソッドは待機する専用の救援信号実行タスクを使用する、上記（３）に記載の装置。
（５）前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時にはより順序正しいシャットダウン・プロシージャを可能にするために、前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用し、前記障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、上記（４）に記載の装置。
（６）前記クラスタ・ノード救援信号発生手段は前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するための機構を含む、上記（１）に記載の装置。
（７）前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの統合部分を含む、上記（１）に記載の装置。
（８）前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、上記（１）に記載の装置。
（９）少なくとも１つのプロセッサと、少なくとも１つのプロセッサに接続されたメモリと、前記メモリ内にあって、クラスタ・ノード装置の障害を表す事前形成された救援メッセージを含むクラスタ・ノード救援信号発生手段と、を含み、前記クラスタ・ノード救援信号発生手段は、現在の実行タスクを使用して前記事前形成された救援メッセージを非同期的に送るためのメソッド及び待機する専用の救援信号実行タスクを使用して前記事前形成された救援メッセージを同期的に送るためのメソッドを含み、前記クラスタ・ノード救援信号発生手段は、障害事象が検出された時に前記クラスタにおける他のノードに前記事前形成された救援メッセージを送る、クラスタ・ノード装置。
（１０）前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時にはより順序正しいシャットダウン・プロシージャを可能にするために前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用し、前記障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、上記（９）に記載のクラスタ・ノード装置。
（１１）前記クラスタ・ノード救援信号発生手段は、前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記クラスタ・ノード装置がそれ自身を前記クラスタから除去することを保証するための機構を含む、上記（９）に記載のクラスタ・ノード装置。
（１２）前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの一部分である、上記（９）に記載のクラスタ・ノード装置。
（１３）前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、上記（９）に記載のクラスタ・ノード装置。
（１４）クラスタにおけるノードが障害事象を経験しようとしていることを表す事前形成された救援メッセージを提供するステップと、差し迫った障害事象が検出された時、前記事前形成された救援メッセージを送るステップと、を含む方法。
（１５）前記事前形成された救援メッセージを処理して送るために専用の救援信号タスクを提供するステップを含む、上記（１４）に記載の方法。
（１６）前記事前形成された救援メッセージを送るステップは、前記差し迫った障害事象が存在する前に十分な時間がある時にはより順序正しいシャットダウン・プロシージャを可能にするために前記事前形成された救援メッセージを非同期的に送るステップ、及び前記差し迫った障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを同期的に送るステップを含む、上記（１４）に記載の方法。
（１７）前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するステップを含む、上記（１４）に記載の方法。
（１８）差し迫った障害事象のタイプを決定するステップを含む、上記（１４）に記載の方法。
（１９）クラスタにおけるノードが障害事象を経験しようとしていることを表す事前形成された救援メッセージ

を供給するステップと、前記事前形成された救援メッセージを処理して送るために専用の救援信号タスクを供給するステップと、差し迫った障害事象を検出するステップと、前記検出された差し迫った障害事象が存在する前に十分な時間がある時には前記事前形成された救援メッセージを前記クラスタにおける他のノードに非同期的に送り、前記検出された差し迫った障害事象が存在する前に十分な時間がない時には前記事前形成された救援メッセージを前記他のノードに同期的に送るステップと、前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するステップと、を含む方法。

（２０）前記事前形成された救援メッセージを非同期的に送るステップは前記専用の救援信号タスクを使用し、前記事前形成されたメッセージを同期的に送るステップは現在の実行タスクを使用する、上記（１９）に記載の方法。

（２１）前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、上記（１９）に記載の方法。

（２２）（Ａ）クラスタにおけるノードの障害を表す事前形成された救援メッセージを含み、障害事象が検出された時に前記クラスタにおける他のノードに前記事前形成された救援メッセージを送るクラスタ・ノード救援信号発生手段と、（Ｂ）前記クラスタ・ノード救援信号発生手段を保持する信号保持媒体と、を含むプログラム製品。

（２３）前記信号保持媒体は伝送媒体を含む、上記（２２）に記載のプログラム製品。

（２４）前記信号保持媒体は記録可能な媒体を含む、上記（２２）に記載のプログラム製品。

（２５）前記クラスタ・ノード救援信号発生手段は、障害事象が検出された時に前記事前形成された救援メッセージを送るための待機する専用の救援信号実行タスクを含む、上記（２２）に記載のプログラム製品。

（２６）前記クラスタ・ノード救援信号発生手段は前記事前形成された救援メッセージを非同期的に送るためのメソッド及び前記事前形成された救援信号を同期的に送るためのメソッドを含む、上記（２２）に記載のプログラム製品。

（２７）前記事前形成された救援メッセージを非同期的に送るためのメソッドは現在の実行タスクを使用して前記事前形成された救援メッセージを送り、前記事前形成された救援メッセージを同期的に送るためのメソッドは待機する専用の救援信号実行タスクを使用する、上記（２６）に記載のプログラム製品。

（２８）前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時には前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用してより順序正しいシャットダウン・プロシージャを可能にし、前記障害事象が存在する前に十分な時間がない時、前記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、上記（２７）に記載のプログラム製品。

（２９）前記クラスタ・ノード救援信号発生手段は、前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記ノードがそれ自身を前記クラスタから除去することを保証するための機構を含む、上記（２２）に記載のプログラム製品。

（３０）前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの統合部分を含む、上記（２２）に記載のプログラム製品。

（３１）前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、上記（２２）に記載のプログラム製品。

（３２）（Ａ）クラスタ・ノード装置の障害を表す事前形成された救援メッセージを含み、現在の実行タスクを使用して前記事前形成された救援メッセージを非同期的に送るためのメソッド及び待機する専用の救援信号実行タスクを使用して前記事前形成された救援メッセージを同期的に送るためのメソッドを含み、障害事象が検出された時、前記事前形成された救援メッセージを前記クラスタにおける他のノードに送るクラスタ・ノード救援信号発生手段、及び（Ｂ）前記クラスタ・ノード救援信号発生手段を保持する信号保持媒体と、を含むプログラム製品。

（３３）前記信号保持媒体は伝送媒体を含む、上記（２２）に記載のプログラム製品。

（３４）前記信号保持媒体は記録可能な媒体を含む、上記（２２）に記載のプログラム製品。

（３５）前記クラスタ・ノード救援信号発生手段は、前記障害事象が存在する前に十分な時間がある時には前記事前形成された救援メッセージを非同期的に送るためのメソッドを使用してより順序正しいシャットダウン・プロシージャを可能にし、前記障害事象が存在する前に十分な時間がない時には記事前形成された救援メッセージを同期的に送るためのメソッドを使用する、上記（３２）に記載のプログラム製品。

（３６）前記クラスタ・ノード救援信号発生手段は、前記クラスタ救援信号が前記クラスタにおける他のノードに送られた後に前記クラスタ・ノード装置がそれ自身を前記クラスタから除去することを保証するための機構を含む、上記（３２）に記載のプログラム製品。

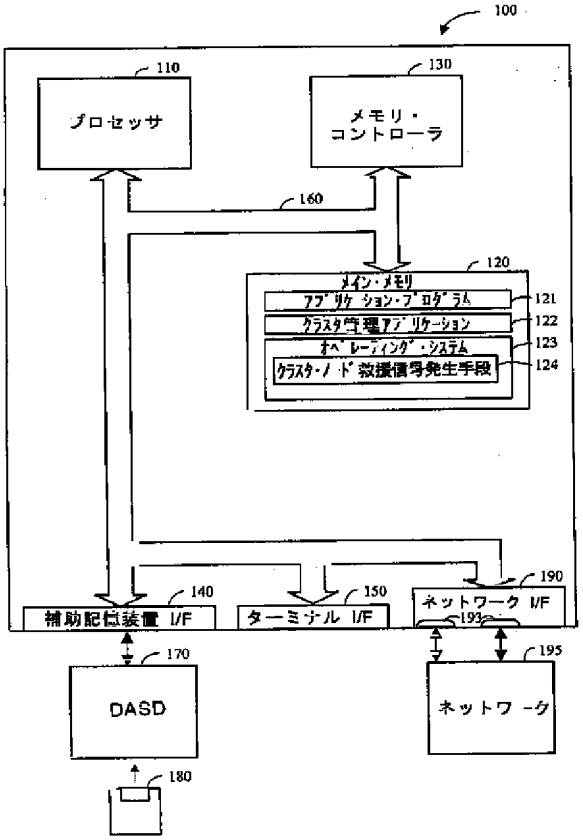
（３７）前記クラスタ・ノード救援信号発生手段は前記メモリにあるオペレーティング・システムの統合部分を含む、上記（３２）に記載のプログラム製品。

（３８）前記事前形成された救援メッセージは予めインスタンス化されたメッセージ・オブジェクトを含む、上記（３２）に記載のプログラム製品。

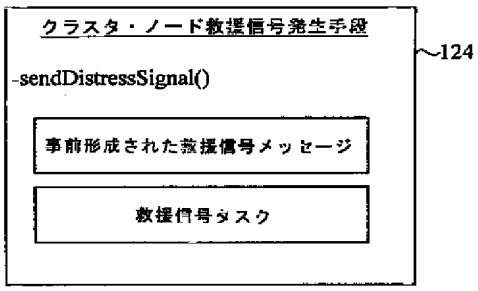
【図面の簡単な説明】

【図１】本発明の望ましい実施例による装置のブロック図である。
【図２】本発明の望ましい実施例に従ってクラスタ・ノード救援信号発生手段を示す概略図である。
【図３】本発明の望ましい実施例によるクラスタ救援信号メソッドの流れ図である。

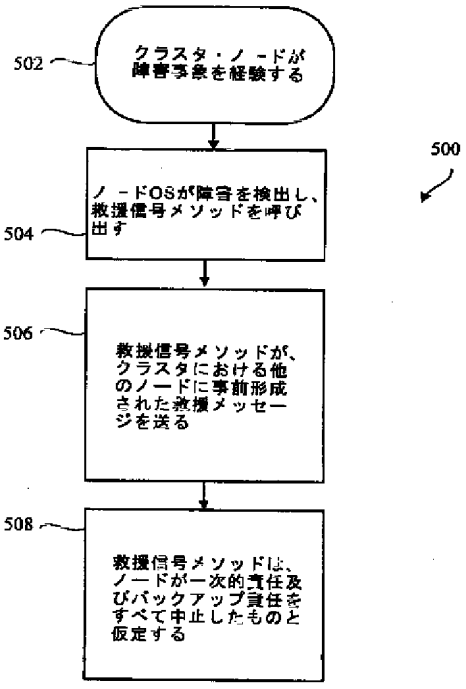
【図１】



【図２】



【図３】



フロントページの続き

(72)発明者　ティモシー・ロイ・ブロック
アメリカ合衆国55901、ミネソタ州、ロチェスタ、エイボン・レーン・エヌ・ダブリュ　4516

(72)発明者　ロドニー・リー・ラブ
アメリカ合衆国55920、ミネソタ州、バイロン、イレブンス・ストリート・エヌ・ダブリュ　421